

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problems Mailbox.**

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-208026

(43)Date of publication of application : 07.08.1998

(51)Int.Cl. G06T 1/00  
G06T 7/00  
G09C 5/00  
H04N 1/387  
H04N 1/44

(21)Application number : 09-336757

(71)Applicant : INTERNATL BUSINESS MACH  
CORP <IBM>

(22)Date of filing : 08.12.1997

(72)Inventor : MINTZER FREDERICK COLE  
YEUNG MINERVA MING-YEE

(30)Priority

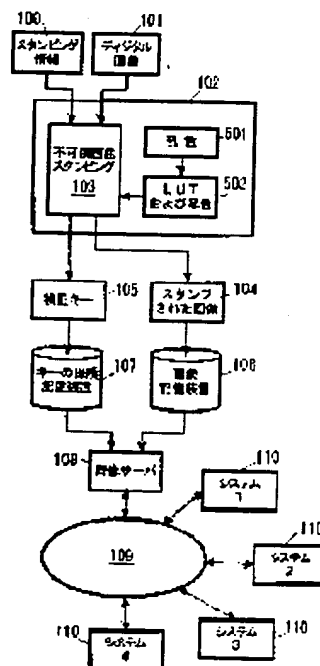
Priority number : 97 780484 Priority date : 08.01.1997 Priority country : US

## (54) PICTURE VERIFICATION DEVICE

(57)Abstract:

PROBLEM TO BE SOLVED: To verify that contents of a picture are not changed after being stamped by discriminating alteration of the stamped picture at the time extracted stamping information is unmatching with source stamping information.

SOLUTION: A picture server 108 verifies the integrity of picture data stored in a picture archive 106. That is, a source picture 104 stamped from the archive is selected, and each information is read into a calculation device after a verification key 105 corresponding to a picture stamped from a key protection storage device 107 is acquired. In the verification processing, the stamped source picture and the verification key are processed to extract stamping information embedded in the picture. Next, extracted stamping information and stamping information 100 known by the server are compared with each other. If they coincide with each other, it is judged that this picture is not changed after being stamped. This verification result is used to confirm the integrity of contents, and a proper processing is performed if it is verified that contents have been changed.



## LEGAL STATUS

[Date of request for examination] 16.10.1998

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-208026

(43) 公開日 平成10年(1998) 8月7日

(51) Int.Cl. <sup>8</sup>	識別記号	F I
G 0 6 T 1/00		G 0 6 F 15/62 3 8 0
7/00		G 0 9 C 5/00
G 0 9 C 5/00		H 0 4 N 1/387
H 0 4 N 1/387		1/44
1/44		G 0 6 F 15/70 4 6 0 A
		審査請求 未請求 請求項の数28 O L (全 23 頁)

(21) 出願番号 特願平9-336757

(22) 出願日 平成9年(1997)12月8日

(31) 優先権主張番号 08/780484

(32) 優先日 1997年1月8日

(33) 優先権主張国 米国 (U S)

(71) 出願人 390009531

インターナショナル・ビジネス・マシー  
ズ・コーポレーション  
INTERNATIONAL BUSIN  
ESS MACHINES CORPO  
RATION

アメリカ合衆国10504、ニューヨーク州  
アーモンク (番地なし)

(72) 発明者 フレデリック・コール・ミンツァー  
アメリカ合衆国10588 ニューヨーク州シ  
ュラブ・オーク オリオール・コート  
3798

(74) 代理人 弁理士 坂口 博 (外1名)

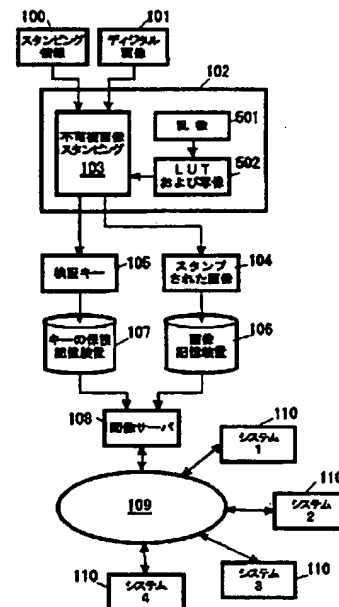
最終頁に続く

(54) 【発明の名称】 画像検証装置

(57) 【要約】

【課題】 本発明の目的は、以前に画像の内容がスタン  
プされた時に降に画像の内容が変更されていないことを  
すばやく検証するシステムを提供することである。

【解決手段】 本発明のシステムは、スタンピング情  
報をソース画像に埋め込み、検証キーを作るスタンピング  
処理と、検証キーに基づいてスタンプされたソース画像  
からスタンピング情報を抽出する検証処理からなる。さ  
らに、検証処理では、画像自体が作られ、この画像か  
ら、視覚的にまたはコンピュータその他の表示装置の使  
用によって、検証を簡単に判定できる。検証処理では、  
画像の変更を検出でき、その位置を限定することができる。  
画像スタンピング処理には、さらに、誤差拡散処理  
が含まれ、その結果、元の画像にスタンピング情報を組  
み合わせた影響を、簡単には知覚できなくなる。画像  
は、悪意のある操作から保護され、所有権は、画像内容  
の完全性を維持することによって保護される。



## 【特許請求の範囲】

【請求項1】(a)少なくとも1つの参照テーブル(LUT)に対応する種値及び写像関数から検証キーを形成することと、

(b)それぞれの抽出値を形成するため、写像関数と少なくとも1つのLUTとに従って、複数のソース画像値のそれぞれを写像することと、

(c)スタンピング情報のうちの対応する部分と同等の変更された抽出値を有する、それぞれの変更されたソース画像値を作るため、スタンピング情報のうちの対応する部分と一致しない抽出値を有する、複数のソース画像値のうちの選択されたソース画像値を変更することによって、スタンプされた画像を作るための画像スタンピング手段と、

スタンプされた画像を受け取り、写像関数に従ってスタンピング情報を抽出するために検証キーを使用し、抽出されたスタンピング情報を提供するための、画像抽出手段と、

スタンピング情報と抽出されたスタンピング情報とが実質的に同等でない場合に、スタンプされた画像への変造があることの信号を送出するための信号送出手段とを含む、複数のソース画像値を有するソース画像を、スタンピング情報を用いてスタンプし、検証するための装置。

【請求項2】検証キーが、種値を含み、画像スタンピング手段が、さらに、乱数ジェネレータ及び種値を使用して少なくとも1つのLUTの各項目を生成するためのテーブル生成手段を含むことを特徴とする、請求項1に記載の画像検証装置。

【請求項3】スタンピング情報が、透かし画像であることを特徴とする、請求項1に記載の画像検証装置。

【請求項4】検証キーとスタンプされた画像を組み合わせるための組み合わせ手段をさらに含み、画像抽出手段が、スタンプされた画像から検証キーを抽出する手段をさらに含むことを特徴とする、請求項1に記載の画像検証装置。

【請求項5】ソース画像が、複数のソース画素値から形成され、透かし画像が、複数の透かし画像画素値から形成されることを特徴とする、請求項3に記載の画像検証装置。

【請求項6】画像スタンピング手段が、さらに、複数のソース画素値のそれぞれをアドレッシングする順序を決定するための手段と誤差拡散手段とを含み、誤差拡散手段が、

1ソース画素と、ソース画像内の、誤差値を含む対応する修正画素とを識別するための手段と、

誤差値を含む修正画素の値と、スタンプされたソース画像内の対応する画素の値との間の差として定義される修正画素誤差の量を決定するための手段と、

隣接画素値の調整の総量が修正画素誤差と同等になるように、連続する隣接ソース画素値を、識別されたソース

画素から離れるにつれて量が減少する修正画素誤差を用いて調整する手段を提供することによって、修正画素誤差を誤差拡散するための手段とを含むことを特徴とする、請求項5に記載の画像検証装置。

【請求項7】少なくとも1つの参照テーブル(LUT)に対応する種値及び写像関数から検証キーを形成するための形成手段と、

それぞれの抽出値を形成するため、写像関数及び少なくとも1つのLUTに従って、複数のソース画像値のそれぞれを写像するための写像手段と、

スタンピング情報のうちの対応する部分と同等の変更された抽出値を有する、それぞれの変更されたソース画像値を作るため、スタンピング情報のうちの対応する部分と一致しない抽出値を有する、複数のソース画像値のうちの選択されたソース画像値を変更するための画素変換手段とを含む、複数のソース画像値を有するソース画像を、スタンピング情報を用いてスタンプするためのシステム。

【請求項8】複数のソース画像値が、複数のソース画素値であり、スタンピング情報が、複数の透かし画像画素値から形成される透かし画像であることを特徴とする、請求項7に記載の画像スタンピング・システム。

【請求項9】信号送出手段が、さらに、スタンプされる情報及び抽出されたスタンピング情報を記憶するための手段と、

比較情報を作るため、スタンプされる情報と抽出されたスタンプされた情報を比較するための手段と、

抽出されたソース画像に対する変造を比較情報から検出し、抽出されたソース画像のうち、変造に対応する少なくとも1つの領域を示すための手段とを含む、請求項1に記載の画像検証装置。

【請求項10】スタンプされた画像が、第1メモリに記憶された複数のスタンプされた画像のうちの1つであり、複数のスタンプされた画像のそれぞれが、対応する検証キーを有し、検証キーのそれぞれが、第2メモリに記憶された複数のキーのうちの1つであることを特徴とする、請求項1に記載の画像検証装置。

【請求項11】第1メモリ内の複数のスタンプされた画像からスタンプされた画像を選択するための手段と、

第2メモリに記憶された複数のキーから、スタンプされた画像に対応する検証キーを取り出すための手段とをさらに含む、請求項10に記載の画像検証装置。

【請求項12】さらに、第1メモリ内の複数のスタンプされた画像のそれぞれの完全性を検証するための画像完全性検証手段を含み、画像完全性検証手段が、(1)複数のスタンプされた画像のうちのそれぞれ及び対応する検証キーを選択することと、(2)対応する検証キーに基づいてスタンプされた画像からスタンピング情報を抽出することと、(3)検証キーを用いてスタンピング情報を検証することによってスタンプされた画像の完全性

を検証することとを反復することとを特徴とする、請求項11に記載の画像検証装置。

【請求項13】(a)少なくとも1つの参照テーブル(LUT)に対応する種値及び写像関数から検証キーを形成することと、

(b)それぞれの抽出値を形成するため、写像関数と少なくとも1つのLUTとに従って、複数の周波数領域係数値のそれぞれを写像することと、

(c)スタンピング情報のうちの対応する部分と同等の変更された抽出値を有する、それぞれの変更された周波数領域係数値を作るため、スタンピング情報のうちの対応する部分と一致しない抽出値を有する、周波数領域係数値のうちの選択された周波数領域係数値を変更することによって、スタンプされた画像を作るための画像スタンピング手段と、

スタンプされた画像を受け取り、写像関数及びLUTに従って複数の周波数領域係数値の一部からスタンピング情報を抽出するために検証キーを使用し、抽出されたスタンピング情報を提供するための、画像抽出手段と、スタンピング情報と抽出されたスタンピング情報とが実質的に同等でない場合に、抽出されたソース画像への変造があることの信号を送出するための信号送出手段とを含む、複数の周波数領域係数値によって表されるソース画像を、スタンピング情報を用いてスタンプし、検証するためのシステム。

【請求項14】複数の周波数領域係数値が、複数のソース画像画素値によって表されるソース画像に対する周波数領域変換演算によって形成されることを特徴とする、請求項13に記載の画像検証システム。

【請求項15】ソース画像画素値に対する変換演算が、離散コサイン変換(DCT)であり、複数の周波数領域係数値の一部が、DCT変換された複数のソース画像画素値の直流係数を含むことを特徴とする、請求項14に記載の画像検証システム。

【請求項16】スタンピング情報が、透かし画像であることを特徴とする、請求項13に記載の画像検証システム。

【請求項17】少なくとも1つの参照テーブル(LUT)に対応する種値及び写像関数から検証キーを生成するためのスタンピング手段と、

対応する抽出値を形成するため、写像関数と少なくとも1つのLUTとに従って、複数の周波数領域係数値の一部のそれぞれを写像するための写像手段と、

スタンピング情報のうちの対応する部分と同等の修正された抽出値を有する、それぞれの修正された周波数領域係数値を作るため、スタンピング情報のうちの対応する部分と一致しない抽出値を有する、複数の周波数領域係数値の一部のうちの選択された一部を修正するための手段とを含む、複数の周波数領域係数値によって表されるソース画像を、スタンピング情報を用いてスタンプする

ためのシステム。

【請求項18】スタンピング情報が、複数の透かし画像値として形成される透かし画像であることを特徴とする、請求項17に記載の画像スタンピング・システム。

【請求項19】複数の周波数領域係数値が、ソース画像の離散コサイン変換(DCT)係数値の組であり、複数の周波数領域係数値の一部が、DCT係数値の組のうちの直流係数を含むことを特徴とする、請求項18に記載の画像スタンピング・システム。

10 【請求項20】(a)少なくとも1つの参照テーブル(LUT)に対応する種値及び写像関数から検証キーを生成するステップと、

(b)それぞれの抽出値を形成するため、写像関数及び少なくとも1つのLUTに従って、複数のソース画像値のそれぞれを写像するステップと、

(c)スタンピング情報のうちの対応する部分と同等の変更された抽出値を有する、それぞれの変更されたソース画像値を作るため、スタンピング情報のうちの対応する部分と一致しない抽出値を有する、複数のソース画像値のうちの選択されたソース画像値を変更するステップと、

20 (d)検証キーによって定義される写像関数及び少なくとも1つのLUTに従って、スタンピング情報を抽出するステップと、

(e)抽出されたスタンピング情報を提供するステップと、

(f)スタンピング情報と抽出されたスタンピング情報とが実質的に同等でない場合に、抽出されたソース画像への変造があることの信号を送出するステップとを含む、複数のソース画像値を有するソース画像を、スタンピング情報を用いてスタンプし、検証する方法。

30 【請求項21】(a)少なくとも1つの参照テーブル(LUT)に対応する種値及び写像関数から検証キーを生成するステップと、

(b)それぞれの抽出値を形成するため、写像関数及びLUTに従って、複数のソース画像値のそれぞれを写像するステップと、

(c)スタンピング情報のうちの対応する部分と同等の修正された抽出値を有する、それぞれの修正されたソース画像値を作るため、スタンピング情報のうちの対応する部分と一致しない抽出値を有する、複数のソース画像値のうちの選択されたソース画像値を修正するステップとを含む、複数のソース画像値を有するソース画像を、スタンピング情報を用いてスタンプする方法。

40 【請求項22】複数のソース画像値が、複数のソース画像画素値であり、スタンピング情報が、複数の透かし画像画素値として形成される透かし画像であり、写像のステップ(a)が、さらに、

50 (a)(1)写像関数と、少なくとも1つのLUTと、複数の透かし画像画素値のうちの対応する1つに従っ

て、スタンプされた画像値のうちの対応する1つを各ソース画素値にセットすることによって、ソース画像に透かし画像を埋め込むステップを含むことを特徴とする、請求項21に記載の画像スタンプ方法。

【請求項23】(a) 1ソース画素と、ソース画像内の、誤差値を含む対応する修正画素とを識別するステップと、

(b) 誤差値を含む修正画素の値と、スタンプされたソース画像内の対応する画素の値との間の差として定義される修正画素誤差の量を決定するステップと、

(c) 隣接画素値の調整の総量が修正画素誤差と同等になるように、各隣接ソース画素の対応する修正画素が変更され、修正画素値が、スタンプ情報を組み込まれ、隣接する変更される画素値が、所望のスタンプ情報をまた組み込まれておらず、後続のスタンプ処理で適当なスタンプ情報を組み込むためにさらに修正されることを特徴とする、連続する隣接ソース画素値を、識別されたソース画素から離れるにつれて量が減少する修正画素誤差を用いて調整することによって、修正画素誤差を誤差拡散するステップとを含む、スタンプされた画像が、複数のソース画素値によって定義され、対応する修正画素を形成するためにそれぞれがスタンプ情報を用いてスタンプされるソース画像であることを特徴とする、誤差拡散されたスタンプされた画像を提供する方法。

【請求項24】(a) 誤差値を含む1ソース画素値と、スタンプされた画像内の対応する修正画素値とを識別するステップと、

(b) 誤差値を含むソース画素値と、対応する修正画素値との間の差として定義される修正画素誤差の量を決定するステップと、

(c) (i) 変更される修正画素のそれぞれと(ii) 修正画素とが、修正画素値としてスタンプ情報のうちの実質的に同様の部分を有するように、各隣接ソース画素の対応する修正画素を変更することによって、連続する隣接ソース画素値を、識別されたソース画素から離れるにつれて量が減少する修正画素誤差を用いて調整することによって、修正画素誤差を誤差拡散するステップとを含む、スタンプされた画像が、複数のソース画素値によって定義され、対応する修正画素を形成するためにそれぞれがスタンプ情報を用いてスタンプされるソース画像であることを特徴とする、誤差拡散されたスタンプされた画像を提供する方法。

【請求項25】(d) 誤差拡散されたスタンプされた画像を提供するため、全ソース画素が識別され、誤差拡散されるまで、ステップ(a)ないし(d)を反復するステップをさらに含むことを特徴とする、請求項24に記載の誤差拡散されたスタンプされた画像を提供する方法。

【請求項26】検証キーに基づいて少なくとも1つの参

照テーブル(LUT)を生成するための手段と、

少なくとも1つのLUTに対応する写像関数を各ソース画像画素に適用し、写像関数の出力値が複数のスタンプ値のうちの対応する1つに対応するように各ソース画素を変更するための手段とを含む、ソース画像が複数のソース画像画素によって表され、スタンプ情報が複数のスタンプ値であることを特徴とする、ソース画像をスタンプ情報を用いてスタンプするためのシステム。

10 【請求項27】検証キーに基づいて少なくとも1つの参照テーブル(LUT)を生成するためのテーブル生成手段と、

写像関数の出力値が複数のスタンプ値のうちの対応する1つを抽出するように、少なくとも1つのLUTに対応する写像関数を各スタンプされたソース画像画素に適用するための手段とを含む、スタンプされたソース画像が複数のスタンプされたソース画像画素によって表され、スタンプ情報が複数のスタンプ値であることを特徴とする、スタンプ情報を有するスタンプされたソース画像からスタンプ情報を抽出するためのシステム。

【請求項28】検証キーが、種値を含み、テーブル生成手段が、種値を使用する乱数ジェネレータを使用することによって、少なくとも1つのLUTの各項目を生成することを特徴とする、請求項26に記載のスタンプされたソース画像からスタンプ情報を抽出するためのシステム。

【発明の詳細な説明】

【0001】

30 【発明の属する技術分野】本発明は、デジタル画像処理の分野に関し、具体的には、マルチメディア・オブジェクト・セキュリティ及び画像の透かし(watermarking)の分野に関する。

【0002】

【従来の技術及び発明が解決しようとする課題】認証と保護を目的とするメモリに記憶されたデジタル画像の完全性の検証及び保護を含むデジタル画像の検証及び保護の分野は、最近重要になり、広く認識されてきた。

40 【0003】この分野では、複数の領域で研究が開始されている。研究領域の1つでは、「信頼に値するデジタル・カメラ」に焦点を合わせ、透かし方式を画像に組み込んで、デジタル写真が修正されたかどうかを判定する。この透かし処理には、既存の公開鍵暗号技術の使用が含まれる。もう1つの研究領域では、最下位ビット(LSB)のビット・プレーン操作による画像への検出不能なデジタル署名のコーディングならびにmシーケンスを使用する画像データへの透かしの線形加算の適用が研究されている。もう1つの技法では、統計的手法を使用してホスト画像に1ビットのデータを埋め込み(パッチワーク)、ランダムなテクスチャのブロックを使用

して類似したテクスチャの領域を置換して、自己相関測定によって形状が回復された、テクスチャを有する領域の同一の対を作成する(テクスチャ・ブロック・コーディング)ことによって、画像に単純なデータを隠す。また、他の研究には、ランダムに生成されたアドレスの組に「デジタル封印」と称する2進ビットの長い列を埋め込み、列内の対応するビットと一致するようにランダムに生成されたアドレスの組の画素のLSBを変更する技法が含まれる。

【0004】上の技法の多くでは、情報が、画像の画素値のLSBに「スタンプ」される。このスタンプ技法は、画像内の視覚的なアーチファクトを発生する可能性がほとんどない。画像を変更する場合、検証によってその修正を判定できるようにするために、LSBが変更される可能性が最も高い。しかし、そのようなLSB操作は、悪意のある攻撃に対して安全ではない。すべての画素値のLSBを変更せずに画像の内容を変更するシステムを作成することは、比較的容易であり、実際、画像全体を置換することができるが、すべての画素のLSBが元のソース画像と同一に保たれる限り、検証処理でそのような変更を検出することはできない。多数の既存の技法のもう1つの短所は、検証処理で修正の範囲を判定できないことである。検証処理は、画像が修正されたかどうかを判定することはできるが、変更が行われた場所を突き止めることはできない。このような情報は、よりよいセキュリティの尺度のために貴重になる可能性がある。

【0005】極端な場合、スタンプ処理は、スタンプ処理自体は何もなく、画像にスタンプされる情報(「スタンプ情報」)が画像そのものになる。この場合、検証処理は、やはり何も行わず、抽出されるスタンプ情報は、スタンプされたソース画像自体になる。スタンプ情報と抽出されたスタンプ情報の最終的な比較は、効果的に、スタンプの時点で既知のソース画像と、検証の時点で既知のソース画像との比較になる。しかし、このスタンプ情報は非常に大きく、情報を記憶するのにかなりのメモリを必要とするので、この比較は非常に効率的ではない。同様に、このスタンプ情報の伝送にはかなりの帯域幅が必要である。この例から、検証処理の望ましい特性の1つが、スタンプ情報の量が非常に少ないことであることがわかる。

【0006】その一方で、もう1つのスタンプ処理は、画像内の所定の位置から画素を抽出し、その画素と画素位置をスタンプ情報として使用することである。この場合、検証処理では、スタンプされた画像内でこの画素が変更されていないかどうかを判定する。この例では、スタンプ情報の量は少ないが、正しい検証が行われる確実性も低い。というのは、単一の所与の画素に影響しない、多数の異なる画像の変更が存在するか

らである。したがって、検証処理の第2の望ましい特性は、検証処理によって、高い確度で、スタンプ以降にスタンプされた画像が修正されたかどうかを判定できることである。

【0007】画像検証のほかに、デジタル透かしは、著作権保護手段のために提案されてきた。たとえば、米国特許第5530759号明細書には、画像の無許可使用を防止するため、画像に可視の「透かし」を配置するシステムが開示されている。米国特許第5488664号明細書には、黒画素と白画素を使用して符号化される印刷された暗号「透かし」を使用して、無許可のアクセス及び修正から視覚情報を保護するための方法及び装置が開示されている。これらは、「可視透かし」の使用の例である。透かし自体は、スタンプされた画像に明示的に表示され、知覚的に容易に目につくようになってい

る。

【0008】

【課題を解決するための手段】画像検証処理で、前に画像の内容にスタンプされた時以降に画像の内容が変更されていないことを検証する。画像検証処理では、ソース画像とスタンプ情報を受け取り、スタンプ情報をソース画像に埋め込んで、スタンプされた画像を作る。キーを用いて、スタンプされた画像からスタンプ情報を抽出できるようにする。埋込み処理の後に、画像検証処理で、キーに基づいてスタンプされた画像からスタンプ情報を抽出し、元のスタンプ情報が抽出されたスタンプ情報と一致しない場合には、スタンプされた画像が変造されていると判定する。

【0009】

【発明の実施の形態】

概要

本発明は、画像内容検証のため画像にスタンプされる「不可視透かし」を介して画像を検証するためのシステム及び方法に関する。透かしは、スタンプ情報からなり、画像スタンプ処理では、透かしの可視の痕跡または画像内の視覚的アーチファクトを全く生成せずに透かし値とソース画像値を組み合わせる。言い換えると、この透かしは知覚的に不可視である。不可視であるから、画像に対する修正で透かしが必ず変更されるが、可視の特徴とは異なり、攻撃者は、画像にスタンプされた変更された不可視のマークを復元できない。したがって、真正性について画像の内容を検証できる。さらに、挿入される透かしは、不可視ではあるが、後程抽出して、所有権情報を示すことができる。

【0010】本発明のシステムは、前に画像の内容にスタンプされた時以降に画像の内容が変更されていないことをすばやく検証する。このシステムは、スタンプされた画像を作るために定義された写像処理に基づいて、スタンプ情報と称するデジタル情報をソース画像に埋め込むスタンプ処理からなる。写像処理は、復号

「キー」によって識別され、この復号キーを用いると、ユーザは、スタンプされた画像からスタンプ情報（復号で、したがって、このキーを検証キーと称する。このシステムには、検証キーによって識別された写像処理に基づいて、スタンプされたソース画像からスタンプ情報を抽出する検証処理も含まれる。本発明の1実施例では、スタンプ情報は、透かし画像（watermark image）と称する所定の画像を定義する情報でもある。この透かし画像を用いると、検証処理で抽出されたスタンプ情報を、スタンプ処理で適用されたスタンプ情報と視覚的に比較できるようになる。もう1つの実施例では、この比較を、コンピュータ・アルゴリズムによって実行できる。最後に、スタンプ情報は、スタンプされた画像の正しい所有権を示すために表示できる。

【0011】本発明は、スタンプされた画像からスタンプ情報を抽出し、画像がスタンプされた時に追加された情報と比較することによって、画像の内容が変更されていないことをすばやく検証する。一致が得られる場合、画像は、スタンプ以降に変更されていないと判定される。本発明には、ソース画像に知覚可能な変化を生じることなく、ソース画像の画像値にスタンプ情報を埋め込むためのシステムも含まれる。したがって、スタンプ情報は、正しい検証キーを用いて簡単かつ安全に検証目的のために抽出でき、検証キーがなければ、スタンプ情報は取得できない。

【0012】本発明のこの実施例では、ソース画像画素値にスタンプ情報を埋め込む処理によって、誤差が導入され、画素値が変更される。しかし、導入される誤差は、誤差拡散処理によって局所的に分散されるので、画像の視覚的な品質低下を引き起こさない。さらに、スタンプ情報は、ソース画素値と導入される誤差の組み合わせによって埋め込まれ、この情報は、最下位ビット（LSB）より上位のさまざまなビット値に隠され、したがって、本発明は、LSB操作の短所をこうむらず、攻撃に対してより安全になっている。さらに、本発明は、検証処理で画像変更の領域を判定し、その位置を限定することができる。

【0013】本発明のスタンプ情報は、任意のデジタル・ビット・ストリームの形とすることができる。好ましい実施例では、商標及びグラフィック記号の形で所有権情報を明瞭に描写する画像を、スタンプ情報として使用する。したがって、本発明では、スタンプ情報も画像であり、これを「透かし画像」と称する。以下の説明では、透かし画像という単語は、スタンプ情報の1形式と同等であり、したがって、これらは交換可能である。

【0014】a) 不可視画像スタンプ処理の要約  
不可視画像スタンプ処理では、透かし画像 $W(I, J)$ をソース画像 $S(I, J)$ に埋め込んで、スタンプ

されたソース画像 $SS(I, J)$ を作る（ $I$ 及び $J$ は、たとえば1列1行を表す、行列内の特定の値を示す整数である）。ソース画像内の各画素は、順番に処理される。この処理では、選択された画素に透かし抽出関数 $WX(*)$ を適用し、抽出された透かし値をテストして、埋め込まれる透かしの値と等しいかどうかを判定する。これらが等しい場合、処理は次の画素に進む。等しくない場合には、抽出された透かしの値が埋め込まれる透かしの値と等しくなるまで選択された画素の値が変更され、これを行うのに必要な変更が計算され、その変更と反対の変更が、誤差拡散を使用して、まだ処理されていない画素に伝播される。ソース画像内のすべての画素がスタンプされるまで、この処理を繰り返す。最終的な産物としてのスタンプされた画像と共に、検証キーが作られる。

【0015】b) 画像検証処理の要約

画像検証処理では、透かし画像 $EW(I, J)$ が、スタンプされたソース画像 $SS(I, J)$ から抽出される。この抽出は、検証キーから透かし抽出関数 $WX(*)$ を計算することによって開始され、その関数をすべての画素 $SS(I, J)$ に適用して、透かし画素 $EW(I, J)$ が作られる。スタンプされたソース画像のすべての画素が処理されるまで、透かし抽出処理を繰り返す。その結果は、抽出された透かし画像である。この画像は、スタンプされた画像の変更及び相違に関する検査のために、視覚的または数値的に元の透かし画像と比較することができる。

【0016】c) J P E G圧縮画像に対する画像検証とスタンプ

高解像度のデジタル画像は、かなりの量の記憶空間を必要とする可能性があり、伝送コストが高くなる可能性がある。これらの画像の多くは、J P E G圧縮規格に従う圧縮形式で配布される可能性が高い。埋込みスタンプ情報または透かしを有する画像を直接圧縮すると、圧縮技法に固有の量子化処理で画素値のほとんどが変更されるので、透かしのほとんどが破壊される。本発明の好ましい実施例では、J P E G圧縮データ形式にコード化された画像に対して不可視画像スタンプ処理と画像検証処理を実行するための方式も提供される。スタンプ情報は、画像を完全に伸長することなく、既存のJ P E G圧縮画像データ・ストリームに直接埋め込むことができる。同様に、画像検証は、J P E G圧縮画像から直接行うことができ、圧縮画像を完全に伸長してから内容を検証する必要はない。圧縮されたJ P E Gソース画像に対する画像検証及び不可視スタンプの処理は、J P E G画像データ・ストリームの直流係数を抽出して直流係数画像を形成し、その直流係数画像に対して画像検証及びスタンプの処理を適用することによって達成される。その後、スタンプされた直流係数を、標準J P E Gデータ形式に従って、圧縮データ・ストリー



ムに再符号化する。

#### 【0017】誤差拡散

好ましい実施例では、誤差を導入し、その後、導入された誤差を滑らかに局所的に拡散することによって、画像をスタンプする。この拡散は、誤差拡散処理の修正及び適合によって達成される。

【0018】画像の中間調化のための誤差拡散技法は、当技術分野で一般に周知である。1つの技法が、フロイド (Robert Floyd) 及びシュタインバーグ (Louis Steinberg) 著、「An Adaptive Algorithm for Spatial Gray Scale」、1975 SID International Symposium, Digest of Technical Papers, 36~37ページに記載されている。誤差拡散の多数の変形が、ウリクニ (R. Ulrich) 著、「Digital Halftoning」、MIT Press (米国マサチューセッツ州ケンブリッジ) 刊、1987年にも記載されている。

【0019】誤差拡散は、各画素位置での拡散誤差  $e_{i,j}$  を0にすることから始まる。数学的には、これを  $e_{i,j} = 0 \forall i, j$  と表す。入力画素は、初期画素位置または第1画素位置の選択された画素の処理から順番に処理され、この入力画素に対して、拡散誤差が決定され、残りの画素のそれぞれについて、以下の処理ステップが行われる。

1. 入力画素値  $i_{p_{i,j}}$  とその画素位置での拡散誤差の値の和として、修正画素値  $m_{p_{i,j}}$  を計算する。数学的には、これを

$$m_{p_{i,j}} = i_{p_{i,j}} + e_{i,j}$$

と表す。

2. 出力画素値  $o_{p_{i,j}}$  を、修正画素値に近い可能な出力値  $q_k$  のうちの1つとして選択する。数学的には、これ

$$o_{p_{i,j}} = Q(m_{p_{i,j}})$$

と表す。ただし、 $Q(x)$  は、 $x$  に近い可能な出力値  $q_k$  を1つ選択する関数である。

3. ここで、量子化誤差は

$$d_{i,j} = m_{p_{i,j}} - o_{p_{i,j}}$$

によって表される。

4. まだ処理されていない画素位置では、この画素位置での量子化誤差に比例する量だけ拡散誤差を増分する。

すなわち、 $\sum c_{r,s} = \gamma$  のもとで

$$e_{i+r,j+s} = e_{i+r,j+s} + c_{r,s} d_{i,j}$$

となる。

【0020】さまざまな誤差拡散技法の相違の多くは、上のステップ4で使用される拡散係数  $c_{r,s}$  に関して行われる選択の変形である。たとえば、上に示したフロイド及びシュタインバーグの論文では、この係数が定数であるが、ランダムな可変係数の使用の例が、参照によって本明細書に組み込まれる米国特許第4654721号明細書に記載されている。

#### 【0021】画像検証システム

図1は、本発明の実施例に従う使用に適した、不可視画像スタンプ処理と画像配布処理のためのシステムのブロック図である。不可視画像スタンプ処理のフローが、本発明の実施例に従う使用に適した記憶及び配布システムと共に図示されている。ブロック103の不可視画像スタンプ処理では、ソース画像101とスタンプ情報100を組み合わせて、スタンプされたソース画像104を作る。検証キー105も作られる。スタンプされたソース画像104は、スタンプ情報100を埋め込まれてはいるが、知覚的にソース画像101とほぼ同一に見えなければならない。言い換えると、スタンプ情報は、元の画像値に直接に不可視の形で隠される。このスタンプ処理は、デジタル・コンピュータ102内で実行される。ブロック103に示された不可視スタンプ処理を、以下の節で詳細に説明し、図5、図6及び図7に図示する。

【0022】スタンプされた画像は、画像アーカイブ106に記憶でき、その後、画像サーバ108によって取り出すことができる。検証キーは、保護記憶装置107に記憶される。画像サーバは、スタンプされた画像を制御し、ローカル・エリア・ネットワークまたはインターネットなどの広域ネットワークとすることができるコンピュータ・ネットワーク109を介して、ネットワークに接続された個々のクライアント・コンピュータ・システム110へ、要求時にスタンプされた画像を配布する。画像サーバ108は、検証キーへのアクセスも制御し、特定の画像の正当な権利を有するクライアントに正しいキーを配布する。スタンプ情報は、正しい検証キーを用いて抽出でき、クライアントのコンピュータ・システム内で表示できる。

#### 【0023】a) 画像検証処理

図2は、画像サーバ108での画像検証処理の機能レベルのブロック図である。画像サーバ108は、画像アーカイブ106に記憶された画像データの完全性を検証する。この検証は、まずアーカイブからスタンプされたソース画像104を選択し、キーの保護記憶装置107からスタンプされた画像に対応する検証キー105を取得した後に、この2つの情報を処理のため計算装置に読み取ることによって達成される。検証処理では、スタンプされたソース画像と検証キーをブロック201で処理して、画像に埋め込まれたスタンプ情報を抽出する。次に、抽出されたスタンプ情報202と、サーバに既知のスタンプ情報100を、ブロック203で比較する。一致が得られる時には、この画像はスタンプされた時に降に変更されていないと判定される。その後、この検証の結果をブロック204で使用して、内容の完全性を確認し、内容が変更されていることが検証される場合には適当な処置を行う。サーバの場合、この処置には、可能な不正使用に対するシステム管理の変更ならびに、内容の完全性を調査されている画像のアクセスの制

限を含めることができる。その後、サーバは、ブロック205で現画像インデックスを更新し、検証処理のための次の画像の再帰的な形での選択に進む。

【0024】図3は、クライアント・コンピュータ・システム110での画像検証処理の機能レベルのブロック図である。この検証処理は、図2の検証処理に類似しているが、複数の変更を有する（以下では、同一の機能性を示す図2のブロックの符号を、括弧に囲んで示す）。このシステムは、スタンプされたソース画像104及び対応する検証キー105をサーバから受け取る。次に、スタンプされた画像と検証キーを、ブロック301（201）で処理して、埋め込まれたスタンピング情報を抽出する。

【0025】抽出されたスタンピング情報302（202）は、ブロック303で、コンピュータ表示モニタ上に所有権情報を示すために表示することができる。その後、（透かし画像の形で）表示されたスタンピング情報を、ブロック304で視覚的な検査に使用して、画像が全体的または部分的に変更されているかどうかを判定できる。画像が変更されていない場合、ブロック305で、後程使用するためにスタンピング情報を記憶でき、その結果、最初の抽出の後にブロック305で記憶されたスタンピング情報と、現在抽出されているスタンピング情報をブロック306（203）で比較して、一致するかどうかを調べることによって、受取の後に画像を周期的に検証することが可能になる。この比較でなんらかの相違が見つかる場合、このシステムは、ブロック307（204）で、悪意のある行為に対する保護のために適当な処置を行うことができる。

【0026】この検証処理では、ブロック201及びブロック301で、スタンプされたソース画像と検証キーを処理して、抽出されたスタンピング情報を作る。本明細書では、スタンプされた画像からのスタンピング情報（たとえば透かし画像）の抽出を、「透かし抽出関数」と称する写像関数に関して説明する。しかし、ソース画像にスタンプする処理は、抽出処理に使用される写像関数の逆関数を適用することによって実行される。

【0027】ブロック201及びブロック301に示された抽出処理の好ましい実施例の流れ図を、図4に示す。透かし画像EW(I, J)は、スタンプされたソース画像SS(I, J)から抽出される。この処理はステップ401から開始され、ステップ402で、検証キーから透かし抽出関数WX(\*)を計算し、ステップ403で、画素インデックスI及びJを、最初に処理する画素のインデックスに初期設定する。

【0028】その後、ステップ404で、透かし抽出関数を画素SS(I, J)に適用して、その画素での抽出された透かしEW(I, J)を、次式によって求める。  

$$EW(I, J) = WX(SS(I, J))$$

【0029】その後、ステップ405で、次に処理する

画素に画素インデックスを増分し、ステップ406で、画素インデックスを検査する。スタンプされたソース画像内のすべての画素を処理済みの場合、透かし抽出処理はステップ407で終了する。スタンプされたソース画像にまだ処理されていない画素がある場合、ステップ404で透かし抽出関数を次の画素に適用し、ステップ405で画素インデックスをもう一度更新し、ステップ406でインデックスをもう一度検査して、スタンプされたソース画像内のすべての画素を処理したかどうかを判定する。

【0030】ステップ402の透かし抽出関数の計算と、ステップ404の透かし抽出処理を実施する、本発明の好ましい実施例を、図10に示す。スタンプされたカラー画像SS(I, J)の場合、画像内のすべての画素(I, J)に、それぞれ赤、緑及び青の色の輝度を示す3つの色成分値すなわち、SS<sub>r</sub>(I, J)、SS<sub>g</sub>(I, J)及びSS<sub>b</sub>(I, J)が含まれる。単色のスタンプされた画像の場合、すべての画素(I, J)に、1つの色輝度値すなわちSS(I, J)だけが含まれることが望ましい。透かし抽出関数WX(\*)は、ステップ402で検証キーに基づいて計算される関数であり、形式的には、

$$EW(I, J) = WX(SS_r(I, J), SS_g(I, J), SS_b(I, J))$$

である。ただし、WX(\*)は、入力値（すなわち画素の色輝度値）を2進出力値にすることのできる写像関数とすることができ、この写像関数は、検証キーによって与えられる。

【0031】好ましい実施例では、1画素の色輝度値のそれぞれに異なる写像関数が適用され、単一の色成分のための写像関数のそれぞれを、色成分写像関数と称する。カラー画像の場合、3つの色成分写像関数の組すなわちF<sub>r</sub>(\*)、F<sub>g</sub>(\*)及びF<sub>b</sub>(\*)が必要であり、単色画像の場合、1つの色成分写像関数すなわちF(\*)が必要である。言い換えると、F(\*)、F<sub>r</sub>(\*)、F<sub>g</sub>(\*)及びF<sub>b</sub>(\*)のそれぞれが、入力値（画素輝度値）を2進出力値に写像できる。

【0032】これらの異なる写像関数は、画素値がテーブル項目のインデックスとして働き、テーブル項目によって1ビット値（「0」または「1」）出力が与えられる2進参照テーブル(LUT)の形で実施される。この場合、検証キーは、1組の2進参照テーブル(LUT)（カラー画像の場合は3つのテーブル、単色画像の場合は1つのテーブル）に関係し、この実施例の場合、1組の2進LUTを生成するために既知の巡回関数ジェネレータに供給できる数（本明細書では種と称する）とすることができる。

【0033】この実施例の2進LUTのそれぞれは、次のように生成される。まず、あるインデックスに対して、テーブル項目は「1」または「0」のいずれかにな

り、これはランダムな形で決定される。たとえば、擬似乱数ジェネレータを使用して、乱数分布を有する任意の数としてテーブル項目を生成できる。当技術分野で既知のとおり、これらの擬似乱数ジェネレータでは、「種」と称する入力数を受け取り、ランダムな出力値を再帰的に生成する巡回関数を使用される。本発明の1実施例では、Cプログラミング言語の標準ライブラリの関数サブルーチン `srand()` 及び `rand()` を使用する。乱数出力値のそれぞれについて、テーブルの値に、乱数出力値が偶数の場合は「0」、乱数出力値が奇数の場合は「1」をセットする。しかし、当業者には明白なとおり、ランダムなテーブル項目は、複数の方法で生成でき、たとえば、線形フィードバック・シフト・レジスタによって生成される値、乱数表の値、または他の直交関数シーケンスを使用することによって生成できる。

【0034】ステップ404の透かし抽出ステップを、図10に示す。スタンプされた画像の画素  $SS(I, J)$  1001に対して、色成分のそれぞれの輝度値  $SS_r(I, J)$ 、 $SS_g(I, J)$  及び  $SS_b(I, J)$  が、それぞれ赤、緑及び青の対応する2進参照テーブル(LUT)すなわち  $LUT_r$ 、 $LUT_g$  及び  $LUT_b$  1002へのインデックスとして働く。各インデックスのテーブル項目を、出力として読み取る。色のそれぞれについて、テーブルから1ビット出力(「0」または「1」)すなわち  $V1$ 、 $V2$  及び  $V3$  が得られる。その後、これら3つの出力ビットのXOR(排他的論理和)演算1003を行って、第  $(I, J)$  画素の最終的な所望の透かしビット値  $EW(I, J)$  1005を得る。数学的に言えば、排他的論理和演算子を $\oplus$ によって表すものとして、カラーのスタンプされた画像の場合は  $EW(I, J) = LUT_r(SS_r(I, J)) \oplus LUT_g(SS_g(I, J)) \oplus LUT_b(SS_b(I, J))$  である(単色画像の場合は  $EW(I, J) = LUT(SS(I, J))$ )。たとえば、画像の画素  $SS^*$  が、 $SS_r^* = 134$ 、 $SS_g^* = 255$ 、 $SS_b^* = 255$  の色輝度値を有し、 $LUT_r(134) = 0$ 、 $LUT_g(255) = 0$ 、 $LUT_b(255) = 1$  である場合、 $SS^*$  の抽出される透かし値  $EW^*$  は  $EW^* = 0 \oplus 0 \oplus 1 = 1$  になる。ただし、下記の数1を $\oplus$ で表す。

【数1】

$\oplus$

【0035】上の例では、XOR演算1003に示されるように透かし抽出関数  $WX(*)$  がXOR関数の組み合わせであり、写像関数はLUTによって与えられる。多くの他の可能な写像関数及び写像関数の組み合わせを、透かし抽出関数として実施できる。

【0036】ブロック203及びブロック306に示された、抽出された透かし値  $EW(I, J)$  と透かし画像

値  $W(I, J)$  の比較は、次のように自動的に達成できる。画素  $(I, J)$  のそれぞれについて、 $EW(I, J)$  と  $W(I, J)$  の間の差の絶対値を記録する。2進透かし値の場合、この値は、2つの値が一致する場合は「0」、2つの値が一致しない場合は「1」になる。すべての画素の差の絶対値を加算し、その和が所定の閾値を超える場合には、スタンプされた画像が変更されていると宣言する。他の自動比較アルゴリズムも実施できる。

#### 10 【0037】不可視画像スタンプ処理

不可視画像スタンプ処理の実施例を示す機能レベルのブロック図を、図5に示す。このブロック図は、図1のブロック103の機能的構成要素を詳細に示す図である。処理ステップの流れ図を、図6及び図7に示し、その詳細を後の節で示す。

【0038】一番上のレベルでは、ソース画像511( $S(I, J)$ )に透かし画像510の形でスタンプ情報埋め込んで、可視のアーチファクトをこうむらずにスタンプされた画像513( $SS(I, J)$ )を作る処理は、ブロック502、503、505、506及び509に示された複数の主要な構成要素からなる。

【0039】スタンプされた画像のトリミングされた部分を検証する能力を組み込むため、本発明の1実施例では、スタンプ情報すなわち透かし画像を、ソース画像の少なくとも1つの部分、潜在的には複数の部分に埋め込む。多くの応用分野では、透かし画像のサイズが、ソース画像のサイズと異なり、はるかに小さいことがしばしばであることに注目されたい。したがって、ブロック509で、透かし画像のサイズをソース画像と同一のサイズに変更して、ソース画像のすべての画素に透かし情報を埋め込めるようにしなければならない。本発明の好ましい実施例では、このサイズ変更が、サイズの小さい透かし画像を、水平方向と垂直方向の両方に、ソース画像のサイズまで複写する(複写された透かし画像のうちでソース画像の境界の外側の部分は、トリミングされる)ことによって達成される。サイズを変更された透かし画像は、ソース画像と同一の寸法を有するが、 $W(I, J)$  によって表され、このサイズを変更された画像を、本発明の説明の残り全体を通じて透かし画像として扱う。

【0040】機能ブロック502は、正しいキーがなければ抽出処理が保護されるようするための、画素から透かし値を抽出する関数の組と検証キーの生成を示す。

【0041】ブロック402で説明したように、好ましい実施例の1つでは、検証キーを、2進参照テーブル(LUT)の組とすることができ、また、必要な2進LUTの生成に使用できる数の既知の関数ジェネレータと1つの数とすることができる。キーを作るために、乱数ジェネレータ501を使用して、数のシーケンスをランダムに生成し、その後、この数のそれぞれを、既知の関

数を介してビット値に写像でき、各ビット値は、検証キー512として働く2進LUTの1項目になる。カラーのソース画像の場合は3つの2進LUTが必要であり、単色のソース画像の場合は1つの2進LUTが必要である。もう1つの手法では、乱数ジェネレータ501が、検証キー512として働く種の数を生じ、この数を、必要な2進LUTを再現するのに使用される既知の巡回関数ジェネレータに供給することができる。

【0042】1組の2進参照テーブルがランダムに生成される際に、テーブル項目に同一の値を有する多数の連続した項目が含まれる可能性がある。この状況は、所望の出力ビット値を得るため、画像スタンピング処理でより大きいレベルへの画素値の調整をもたらす、このため、スタンプされた画像に望ましくないアーチファクトが発生する可能性がある。したがって、本発明のもう1つの実施例では、同一の値を有する連続する項目の数を制限して、画素値（したがって、画素の輝度）の潜在的に大きい調整を克服する。本発明のこの実施例では、連続するテーブル項目値の数が小さくなるように制限するために参照テーブルを処理する。本明細書で説明する実施例の場合、許容可能な連続するテーブル項目の最大個数は、4または5である。

【0043】2進LUTは、ブロック503に示される、ソース画像511すなわち $S(I, J)$ の透かし値の計算で、2進写像関数として働く。透かし値の計算または抽出は、機能性においては前の節で詳細に説明した図4のステップ404と同一である。唯一の相違は、入力、スタンプされた画素値ではなくソース画像の画素値 $S(I, J)$ であることである。カラーのソース画像画素 $S(I, J)$ の場合、赤、緑及び青の3つの色成分は、それぞれ $S_r(I, J)$ 、 $S_g(I, J)$ 及び $S_b(I, J)$ によって表され、3つのLUTによって提供される3つの写像関数の組は、それぞれ $LUT_r(\cdot)$ 、 $LUT_g(\cdot)$ 及び $LUT_b(\cdot)$ によって表される。単色画像の場合、画素の輝度値は $S(I, J)$ によって表され、2進LUTによって提供される写像関数は $LUT(\cdot)$ によって表される。透かし抽出関数 $WX(\cdot)$ は、写像関数の関数である。画素 $(I, J)$ のそれぞれについて、抽出される透かし値 $EW(I, J)$ は、②がXOR演算子であるものとして、カラーのソース画像の場合は

$$EW(I, J) = LUT_r(S_r(I, J)) \oplus LUT_g(S_g(I, J)) \oplus LUT_b(S_b(I, J))$$

として計算され、単色画像の場合は、 $EW(I, J) = LUT(S(I, J))$ として計算される。

【0044】その後、ブロック504に示された比較処理で、ソース画像画素の計算された透かし値を、サイズを変更された透かし画像 $W(I, J)$ の所望の値と比較して、これらの値が一致するかどうかを調べる。

【0045】ゲートウェイ507は、2つの部分を有す

る画素修正の反復処理の実施を制御する。第1の部分では、ステップ505で、最初のテストで透かし値の一致が見つからない場合に、計算された透かし値を、透かし画像の対応する画素の所望の値に強制的に一致させるように、ソース画素画素値のそれぞれを変更する。画素変更の場合、修正によって導入される誤差が、スタンプされた画像に可視のアーチファクトを作ってはならないので、誤差は、画像画素輝度値を徐々に変更するために、所定のゆっくり増加するレベルで導入される。したがって、画素修正の処理は、複数回の反復を要する場合がある。変更された画素の透かし値は、新しい誤差調整反復のそれぞれについてブロック508で再計算されて、ブロック504で、再計算された値が所望の値と一致するかどうか検査される。第2の部分では、計算された透かし値が所望の透かし値と一致した後に、反復を停止し、ブロック506で、近接する画素の平均色輝度値を一定に保つ形で、画素値変更ステップで導入された誤差を近接する画素に拡散する。

【0046】好ましい実施例の1つでは、カラー画像の画素の場合に、画素の3つの色の値が変更される。画素値は、ひずみが最小になる形で修正されなければならない。どの色輝度値でも、所与の反復で修正できる。調整のレベルは、+1または-1から開始され、後続の反復で徐々に増加する。各色成分の調整の現在レベルの記録を使用して、レベル変更の大きさが徐々に増加するようにする。変更が画像の視覚的品質の大きな相違に繋がらないように、調整の最大レベルを制御する。

【0047】所与の反復で変更される色の選択は、事前に定義されたテーブルを使用するか、選択処理に乱数ジェネレータを組み込むことによって達成できる。乱数ジェネレータ使用の利点は、修正が、大多数の画素について特定の色に集中するのではなく、3色のすべてに分散することである。この効果は、大きい面積が単一の輝度値になる可能性がある合成画像の符号化で特に顕著である。修正する画素の選択に規則正しい方式を用いると、画像内の特定の色成分に変更が集中する可能性があり、したがって、これは望ましくない。レベル調整の符号（+または-）の選択も、局所的に作られる誤差を滑らかにするために、画素ごとにランダムに行われる。

【0048】単色画像の場合、画素変更は、カラー画素の変更で使用された方法の単純な適応である。画素値の漸進的な調整は、3つの色成分ではなく1つの輝度成分に限られる。

【0049】上で説明した画素値変更によって導入される誤差は、ブロック506に示されるように拡散され、その結果、平均画像輝度値が保存される。これによって、変更を滑らかにするために誤差が画像上で局所的に拡散されるので、正しい平均色が作られる。

【0050】誤差拡散は、真の色から制限付きカラーパレットへの変換の際に、近接する画素の色の变化を滑

らかにするの一般的に使用されている。この処理では、急激な変化の代わりに自然なテクスチャと色の変化を追加することによって、高品質のカラー画像が作られる。誤差拡散は、同様の理由から、画素値の変更によって生じる誤差（色変換処理での量子化誤差と同等）が近接する画素に分散するように、符号化装置で使用される。この方法を用いると、正しい平均色を保持でき、望ましくない局所アーチファクトが除去される。本発明には、本発明の具体的な目的に適合された、前に説明した\*

$$MP(I, J) = S(I, J) + E(I, J) \quad (1)$$

2. 出力画素値  $SS(I, J)$  は、所望の透かし値と一致するビット出力を与える、変更された修正画素値である。

$$D(I, J) = MP(I, J) - SS(I, J) \quad (2)$$

4. まだ処理されていない近接する画素位置で、現在位置での出力誤差に比例する量だけ拡散誤差を調整する。★

$$E(I+R, J+S) \leftarrow E(I+R, J+S) + C(R, S) D(I, J) \quad (3)$$

を行う。式(1)及び式(2)を組み合わせると、次式が☆ ☆得られる。

$$SS(I, J) = MP(I, J) - D(I, J) = S(I, J) + E(I, J) - D(I, J) \quad (4)$$

拡散係数の合計  $\sum_{R,S} C(R, S)$  を1に制限することによって、 $\sum E(I, J) = \sum D(I, J)$  が得られる。すべての  $(I, J)$  に対して(4)を合計すると、次式が得られる。

$$\sum SS(I, J) = \sum S(I, J) \quad (5)$$

【0052】これによって、平均輝度値が保存され、正しい平均色が作られることが保証される。3つの色RG◆

$$0.5, (R, S) = (1, 0), (0, 1) \text{ の場合}$$

$$C(R, S) = \{ \quad (6)$$

0, それ以外

【0054】単色画像の場合、誤差の拡散は、画像の輝度値に限られ、上で説明した方式から簡単に適合できる。

【0055】このスタンピング処理が完了した時に、スタンプされた画像513が、出力として作られ、その画素値にスタンピング情報が組み込まれている。出力のスタンプされた画像513のほかに、検証キー512が、後続の画像検証処理の目的のため、この処理で作られる。

【0056】本発明の画像スタンピング処理の流れ図を、図6に示す。画像スタンピング処理は、ステップ601で開始され、ステップ602で検証キーを計算し、ステップ603で透かし抽出関数すなわち  $WX(\cdot)$  を計算し、ステップ604で、処理される画素のインデックス  $I$  及び  $J$  を初期設定する。その後、ステップ605で、位置  $(I, J)$  の画素を処理する。ステップ605は、誤差の導入によって位置  $(I, J)$  の画素値を修正し、近接する画素に誤差を拡散する反復処理である。このステップを、図7に詳細に示す。ステップ608で、

\* 誤差拡散処理の適合が含まれる。

【0051】 $E(I, J)$  を、画素  $(I, J)$  の所与の色の拡散誤差と定義する。すべての  $(I, J)$  について、 $E(I, J)$  を0に初期設定する。すべての画素  $(I, J)$  について、以下の誤差拡散処理ステップを行う。

1. 現在の修正画素値  $MP(I, J)$  は、ソース画像画素値  $S(I, J)$  と、その画素位置での拡散誤差の値の合計である。すなわち、

※ 3. 出力誤差  $D(I, J)$  は、現在の修正画素値と出力画素値の間の差として計算される。すなわち、

★したがって、いくつかの  $(R, S)$  について、

◆ Bについて、各色の誤差拡散処理は独立に実行される。言い換えると、同一の定式化を、赤(R)、緑(G)及び青(B)の輝度値に適用できる。誤差も、結合された色の誤差拡散処理の変形を使用して拡散できる。

【0053】本発明の好ましい実施例では、以下の拡散係数を使用する。

画素インデックス  $(I, J)$  を次に処理する画素のインデックスに増分し、ステップ609で、ソース画像全体を処理したかどうかを判定するテストを行う。ソース画像全体を処理した場合、画像スタンピング処理はステップ601で完了する。そうでない場合には、この処理はステップ605で次の画素に進む。

【0057】図7は、ステップ605によって説明した処理の流れ図である。ステップ701で、ソース画像画素  $S(I, J)$  と誤差値  $E(I, J)$  を与えられて、修正誤差値  $MP(I, J)$  を計算する。ステップ702で、透かし抽出関数を使用して、画素  $MP(I, J)$  から透かし値を抽出する。その後、ステップ703で、値  $EW$  をテストして、埋め込まれる透かしの値  $W(I, J)$  と等しいかどうかを判定する。

【0058】値  $EW$  と  $W(I, J)$  が一致する場合、ステップ704で、出力画素の値  $SS(I, J)$  に  $S(I, J)$  と等しい値をセットする。位置  $(I, J)$  でのこの画素のスタンピング処理は、ステップ713で完了する。

【0059】値EWとW(1, J)が一致しない場合、この処理は、ステップ705に進んで、 $\Delta$ 計算子を初期設定し、ステップ706で $\Delta$ の新しい値を計算し、ステップ707で変更された修正画素値MSを作成し、ステップ708で、透かし値EMWをMSから抽出する。その後、ステップ709で、EMWをテストして、これがW(1, J)と等しいかどうかを判定する。

【0060】値EMWとW(1, J)が等しい場合、ステップ710で、出力画素の値にMSの値をセットし、ステップ711で、MP(1, J)とMSの間の差を計算する。この差が、ステップ712で拡散される誤差である。位置(1, J)での画像画素のスタンピング処理は、ステップ713で終了する。

【0061】値EMWとW(1, J)が等しくない場合、ステップ706に戻ることに伴って、 $\Delta$ の新しい値を計算し、ステップ707でMSの新しい値を計算し、ステップ708でEMWの新しい値を計算し、ステップ709でEMWとW(1, J)が等しいかどうかを調べる。

【0062】差D(1, J)は、まだ処理されていない近接画素に拡散される誤差である。位置(1, J+1)及び(1+1, J)の近接画素の拡散誤差Eは、 $E(1, J+1) = C(0, 1)D(1, J)$   
 $E(1+1, J) = C(1, 0)D(1, J)$ である。

【0063】好ましい実施例では、C(0, 1)とC(1, 0)は0.5と同等である。以下で、本発明の1実施例の処理の例を説明する。画像画素S\*が、色輝度値

$$(S^*_r, S^*_g, S^*_b) = S^*$$

を有し、これと同一の画素位置での誤差ベクトルE\*が、色輝度値

$$(E^*_r, E^*_g, E^*_b) = E^*$$

を有すると仮定すると、

$$S^*_r = 119, S^*_g = 111, S^*_b = 250$$

$$E^*_r = 1, E^*_g = -1, E^*_b = 0$$

かつ

$$LUT_r(119) = 0, LUT_r(120) = 0, LUT_r(121) = 0$$

$$LUT_g(9) = 0, LUT_g(10) = 0, LUT_g(11) = 1$$

$$LUT_b(249) = 0, LUT_b(250) = 1, LUT_b(251) = 1$$

の場合には、

$$MP^* = S^* + E^* = (120, 10, 250)$$

になり、MP\*で抽出される透かし値は

$$EW^* = WX(MP^*) = 0 \oplus 0 \oplus 1 = 1$$

になる。所望の透かし値W\*が0の場合、すなわち、 $EW^* \neq W^*$ の場合、MP\*を修正する。まず、 $\Delta$ を(0, 0, 0)に初期設定する。次に、新しいデルタを計算す

る。 $\Delta = (1, 0, 0)$ の場合、修正画素値は、

$$MS = MP^* + \Delta = (121, 10, 250)$$

になり、

$$EMW = WX(MS) = 0 \oplus 0 \oplus 1 = 1$$

になる。 $EMW \neq W^*$ なので、反復が継続する。新しい

$\Delta$ を計算し、たとえば $\Delta = (0, 1, 0)$ であれば、

$$MS = MP^* + \Delta = (120, 11, 250)$$

$$EMW = WX(MS) = 0 \oplus 1 \oplus 1 = 0$$

になる。今度はEMWとW\*が一致するので、SS\*にMP\*

の値をセットし、出力誤差D\*、この例では(0, -1, 0)を、近接する画素に拡散する。EMWがW\*と一致しない場合、新しい $\Delta$ を計算し、反復を継続する。

【0064】JPEG圧縮画像に対する画像スタンピング及び検証

本発明の実施例の装置及び方法は、JPEG圧縮画像のマークにも使用できる。圧縮される画像または入力画像のそれぞれは、画素からなると仮定し、各画素は複数の色成分からなる可能性があるとして仮定する。たとえば、各画素は、3つの色成分すなわち赤、緑及び青からなるものとしてすることができる。i行j列の入力画像画素の第C色成分の値を、P(C, i, j)と表す。単一の色成分Cについて、画像P(C, i, j)を、入力画像の第Cカラー・プレーンと称する。ここで、JPEG圧縮を簡単に再検討し、JPEG圧縮画像にスタンピング情報を追加し、その内容を後程検証するための処理ステップを要約する。

【0065】JPEGの基礎の形式では、画像のカラー・プレーンのそれぞれが、8行8列に配置される画素の、重なり合わないブロックに分解される。MがM行目のブロックを表し、NがN列目のブロックを表し、mがブロック内のm行目の画素を表し、nがブロック内のn列目の画素を表し、b(C, M, N, m, n)が第Cカラー・プレーンの(M, N)番目のブロックの(m, n)番目の画素であるとして、 $0 < m < 9, 0 < n < 9$ のm, nについて、 $b(C, M, N, m, n) = P(C, 8 \cdot M + m, 8 \cdot N + n)$ である。

【0066】次に、JPEGでは、離散コサイン変換を使用して、ブロック内の64個の値を変換する。すなわち、 $8 \times 8$ のブロックに配置された64個の変換係数を作る。離散コサイン変換の後に、JPEG圧縮では、係数のそれぞれをスカラーで除算し、その結果を整数値に量子化する処理ステップが必要である。その後のJPEG圧縮ステップでは、量子化された変換係数を、ハフマン・コーディングを使用してコード化する。

【0067】ブロックb(C, M, N, m, n)の変換によって作られる離散コサイン変換係数のブロックをB(C, M, N, p, q)と表すと、そのブロックの直流係数B(C, M, N, 0, 0)は、第Cカラー・プレー

ンの(M, N)番目のブロックの64個の値の平均である。この直流係数から、直流係数画像BD(C, M, N)を次のように定義する。

$$BD(C, M, N) = B(C, M, N, 0, 0)$$

【0068】直流係数画像は、各次元で入力画像のサイズの1/8である。サンプルの画像を図11に示し、その緑カラー・プレーンから作られた直流係数画像を図12に示す。

【0069】本発明の方法を用いてJPEG画像にマークを付けるため、本発明の実施例では、JPEGによる直流係数値の計算の後に、スタンピング情報を挿入し、処理を実行する。この処理では、この方法で非圧縮画像にマークするのと正確に同一の形で1つまたは複数の直流係数画像にマークする。このマーキングの後に、JPEG処理は、通常通りコサイン変換係数の量子化とエン트로ピ・コーディングに進む。

【0070】1実施例によれば、同一の位置(M, N)での複数の直流係数画像の複数の直流係数値を、多色直流係数画像の単一画素に関連する複数の色値として扱う。この場合、マーキングは、同一の数の色を有する非圧縮多色画像のマーキングと同一である。他の実施例では、他のカラー・プレーンの直流係数画像と独立に、1つまたは複数の直流係数画像にマークを付ける。

【0071】図8は、JPEG圧縮形式のソース画像に対して不可視画像スタンピング処理を達成するための好ましい実施例を示す、機能レベルのブロック図である。ソース画像801について、この画像に、JPEG圧縮802をかけ、JPEG圧縮画像803と称するJPEG標準規格に適合するデータ・ビット・ストリームに符号化することができる。JPEG圧縮画像803から、ブロック804で、そのビット・ストリームから直接に直流係数を取得する。直流係数は、JPEG標準規格では交流係数とは異なる形で別々にコード化される。直流係数画像805は、抽出された直流係数を組み合わせることによって形成される。直流画像は、元の画像のアイコン版であり、JPEG圧縮で8×8ブロックを使用する場合には、各次元で1/8のサイズである。その後、この直流画像を、直流係数画像と同一サイズのスタンピング透かし画像806と共に、前述の画像スタンピング処理807(103)への入力ソース画像として扱う。検証キー809も作られる。ステップ808で、スタンピング処理からスタンプされた直流係数画像が作られ、直流画像内の画素値は、透かし情報を埋め込むために変更される。これは、直流係数画像805の直流係数が、それ相応に変更されることを意味する。修正された直流係数は、ブロック810で再コード化され、JPEGデータ・ストリームに統合されて、標準JPEG圧縮画像811が作られる。

【0072】図9は、JPEG圧縮形式のスタンプされた画像に対する画像検証処理を示す、機能レベルのブ

ック図である。入力画像は、直流係数に透かし情報を埋め込まれた標準JPEG圧縮画像901である。このスタンプされた画像のデータ・ストリームは、JPEG標準規格に適合するので、図9のブロック903に類似するブロック902で、直流係数をデータ・ストリームから直接に抽出して、スタンピング情報を含む直流係数画像903を作る。このスタンプされた直流係数画像は、対応する検証キー905と共に、ブロック904の画像検証処理への入力として使用される。

【0073】上の方式について、クロミナンス値のサブサンプリングがあるJPEG方式の変形では、透かし画像の埋込みと検証に使用するための透かし画像の後続の抽出が、輝度画素値だけに限られる。

【0074】上の実施例の主要な望ましい特徴の1つが、JPEG圧縮画像を伸長してから内容を検証する必要なしに、画像の検証及びスタンピングをJPEG圧縮画像に対する処理から直接に達成できることである。もう1つの特徴は、JPEG圧縮の量子化動作でほとんどの画素値が変更される可能性が高く、したがって、埋め込まれた透かし情報の大部分が失われるので、情報をJPEGデータ・ストリームに直接埋め込むことによって、透かし情報がJPEG圧縮処理で失われなくなることである。

【0075】本発明の好ましい実施例を図示し、説明してきたが、これらの実施例は、例としてのみ提供されることを理解されたい。当業者であれば、本発明の趣旨から逸脱することなく、多数の変形、変更及び置換を思い浮かべるであろう。したがって、請求の範囲は、そのような変形のすべてを、本発明の趣旨及び範囲に含まれるものとして含むことが意図されている。

【0076】まとめとして、本発明の構成に関して以下の事項を開示する。

【0077】(1)(a)少なくとも1つの参照テーブル(LUT)に対応する種値及び写像関数から検証キーを形成することと、(b)それぞれの抽出値を形成するため、写像関数と少なくとも1つのLUTとに従って、複数のソース画像値のそれぞれを写像することと、

(c)スタンピング情報のうちの対応する部分と同等の変更された抽出値を有する、それぞれの変更されたソース画像値を作るため、スタンピング情報のうちの対応する部分と一致しない抽出値を有する、複数のソース画像値のうちの選択されたソース画像値を変更することによって、スタンプされた画像を作るための画像スタンピング手段と、スタンプされた画像を受け取り、写像関数に従ってスタンピング情報を抽出するために検証キーを使用し、抽出されたスタンピング情報を提供するための、画像抽出手段と、スタンピング情報と抽出されたスタンピング情報とが実質的に同等でない場合に、スタンプされた画像への変造があることの信号を送出するための信号送出手段とを含む、複数のソース画像値を有する

ソース画像を、スタンピング情報を用いてスタンプし、検証するための装置。

(2) 検証キーが、種値を含み、画像スタンピング手段が、さらに、乱数ジェネレータ及び種値を使用して少なくとも1つのLUTの各項目を生成するためのテーブル生成手段を含むことを特徴とする、上記(1)に記載の画像検証装置。

(3) スタンピング情報が、透かし画像であることを特徴とする、上記(1)に記載の画像検証装置。

(4) 検証キーとスタンプされた画像を組み合わせるための組み合わせ手段をさらに含み、画像抽出手段が、スタンプされた画像から検証キーを抽出する手段をさらに含むことを特徴とする、上記(1)に記載の画像検証装置。

(5) ソース画像が、複数のソース画素値から形成され、透かし画像が、複数の透かし画像画素値から形成されることを特徴とする、上記(3)に記載の画像検証装置。

(6) 画像スタンピング手段が、さらに、複数のソース画素値のそれぞれをアドレッシングする順序を決定するための手段と誤差拡散手段とを含み、誤差拡散手段が、1ソース画素と、ソース画像内の、誤差値を含む対応する修正画素とを識別するための手段と、誤差値を含む修正画素の値と、スタンプされたソース画像内の対応する画素の値との間の差として定義される修正画素誤差の量を決定するための手段と、隣接画素値の調整の総量が修正画素誤差と同等になるように、連続する隣接ソース画素値を、識別されたソース画素から離れるにつれて量が減少する修正画素誤差を用いて調整する手段を提供することによって、修正画素誤差を誤差拡散するための手段とを含むことを特徴とする、上記(5)に記載の画像検証装置。

(7) 少なくとも1つの参照テーブル(LUT)に対応する種値及び写像関数から検証キーを形成するための形成手段と、それぞれの抽出値を形成するため、写像関数及び少なくとも1つのLUTに従って、複数のソース画像値のそれぞれを写像するための写像手段と、スタンピング情報のうちの対応する部分と同等の変更された抽出値を有する、それぞれの変更されたソース画像値を作るため、スタンピング情報のうちの対応する部分と一致しない抽出値を有する、複数のソース画像値のうちの選択されたソース画像値を変更するための画素変更手段とを含む、複数のソース画像値を有するソース画像を、スタンピング情報を用いてスタンプするためのシステム。

(8) 複数のソース画像値が、複数のソース画素値であり、スタンピング情報が、複数の透かし画像画素値から形成される透かし画像であることを特徴とする、上記

(7)に記載の画像スタンピング・システム。

(9) 信号送出手段が、さらに、スタンプされる情報及び抽出されたスタンピング情報を記憶するための手段

と、比較情報を作るため、スタンプされる情報と抽出されたスタンプされた情報を比較するための手段と、抽出されたソース画像に対する変造を比較情報から検出し、抽出されたソース画像のうち、変造に対応する少なくとも1つの領域を示すための手段とを含む、上記(1)に記載の画像検証装置。

(10) スタンプされた画像が、第1メモリに記憶された複数のスタンプされた画像のうちの1つであり、複数のスタンプされた画像のそれぞれが、対応する検証キーを有し、検証キーのそれぞれが、第2メモリに記憶された複数のキーのうちの1つであることを特徴とする、上記(1)に記載の画像検証装置。

(11) 第1メモリ内の複数のスタンプされた画像からスタンプされた画像を選択するための手段と、第2メモリに記憶された複数のキーから、スタンプされた画像に対応する検証キーを取り出すための手段とをさらに含む、上記(10)に記載の画像検証装置。

(12) さらに、第1メモリ内の複数のスタンプされた画像のそれぞれの完全性を検証するための画像完全性検証手段を含み、画像完全性検証手段が、(1)複数のスタンプされた画像のうちのそれぞれ及び対応する検証キーを選択することと、(2)対応する検証キーに基づいてスタンプされた画像からスタンピング情報を抽出することと、(3)検証キーを用いてスタンピング情報を検証することによってスタンプされた画像の完全性を検証することとを反復することを特徴とする、上記(11)に記載の画像検証装置。

(13) (a) 少なくとも1つの参照テーブル(LUT)に対応する種値及び写像関数から検証キーを形成することと、(b)それぞれの抽出値を形成するため、写像関数と少なくとも1つのLUTに従って、複数の周波数領域係数値のそれぞれを写像することと、(c)スタンピング情報のうちの対応する部分と同等の変更された抽出値を有する、それぞれの変更された周波数領域係数値を作るため、スタンピング情報のうちの対応する部分と一致しない抽出値を有する、周波数領域係数値のうちの選択された周波数領域係数値を変更することによって、スタンプされた画像を作るための画像スタンピング手段と、スタンプされた画像を受け取り、写像関数及びLUTに従って複数の周波数領域係数値の一部からスタンピング情報を抽出するために検証キーを使用し、抽出されたスタンピング情報を提供するための、画像抽出手段と、スタンピング情報と抽出されたスタンピング情報とが実質的に同等でない場合に、抽出されたソース画像への変造があることの信号を送出するための信号送出手段とを含む、複数の周波数領域係数値によって表されるソース画像を、スタンピング情報を用いてスタンプし、検証するためのシステム。

(14) 複数の周波数領域係数値が、複数のソース画像画素値によって表されるソース画像に対する周波数領域



変換演算によって形成されることを特徴とする、上記  
(13)に記載の画像検証システム。

(15) ソース画像画素値に対する変換演算が、離散コサイン変換(DCT)であり、複数の周波数領域係数値の一部が、DCT変換された複数のソース画像画素値の直流係数を含むことを特徴とする、上記(14)に記載の画像検証システム。

(16) スタンピング情報が、透かし画像であることを特徴とする、上記(13)に記載の画像検証システム。

(17) 少なくとも1つの参照テーブル(LUT)に対応する種値及び写像関数から検証キーを生成するためのスタンピング手段と、対応する抽出値を形成するため、写像関数と少なくとも1つのLUTに従って、複数の周波数領域係数値の一部のそれぞれを写像するための写像手段と、スタンピング情報のうちの対応する部分と同等の修正された抽出値を有する、それぞれの修正された周波数領域係数値を作るため、スタンピング情報のうちの対応する部分と一致しない抽出値を有する、複数の周波数領域係数値の一部のうちの選択された一部を修正するための手段とを含む、複数の周波数領域係数値によって表されるソース画像を、スタンピング情報を用いてスタンプするためのシステム。

(18) スタンピング情報が、複数の透かし画像値として形成される透かし画像であることを特徴とする、上記(17)に記載の画像スタンピング・システム。

(19) 複数の周波数領域係数値が、ソース画像の離散コサイン変換(DCT)係数値の組であり、複数の周波数領域係数値の一部が、DCT係数値の組のうちの直流係数を含むことを特徴とする、上記(18)に記載の画像スタンピング・システム。

(20) (a) 少なくとも1つの参照テーブル(LUT)に対応する種値及び写像関数から検証キーを生成するステップと、(b) それぞれの抽出値を形成するため、写像関数及び少なくとも1つのLUTに従って、複数のソース画像値のそれぞれを写像するステップと、(c) スタンピング情報のうちの対応する部分と同等の修正された抽出値を有する、それぞれの変更されたソース画像値を作るため、スタンピング情報のうちの対応する部分と一致しない抽出値を有する、複数のソース画像値のうちの選択されたソース画像値を変更するステップと、(d) 検証キーによって定義される写像関数及び少なくとも1つのLUTに従って、スタンピング情報を抽出するステップと、(e) 抽出されたスタンピング情報を提供するステップと、(f) スタンピング情報と抽出されたスタンピング情報とが実質的に同等でない場合に、抽出されたソース画像への変造があることの信号を送出するステップとを含む、複数のソース画像値を有するソース画像を、スタンピング情報を用いてスタンプし、検証する方法。

(21) (a) 少なくとも1つの参照テーブル(LUT

T)に対応する種値及び写像関数から検証キーを生成するステップと、(b) それぞれの抽出値を形成するため、写像関数及びLUTに従って、複数のソース画像値のそれぞれを写像するステップと、(c) スタンピング情報のうちの対応する部分と同等の修正された抽出値を有する、それぞれの修正されたソース画像値を作るため、スタンピング情報のうちの対応する部分と一致しない抽出値を有する、複数のソース画像値のうちの選択されたソース画像値を修正するステップとを含む、複数のソース画像値を有するソース画像を、スタンピング情報を用いてスタンプする方法。

(22) 複数のソース画像値が、複数のソース画素値であり、スタンピング情報が、複数の透かし画像画素値として形成される透かし画像であり、写像のステップ(a)が、さらに、

(a) (1) 写像関数と、少なくとも1つのLUTと、複数の透かし画像画素値のうちの対応する1つに従って、スタンプされた画像値のうちの対応する1つを各ソース画素値にセットすることによって、ソース画像に透かし画像を埋め込むステップを含むことを特徴とする、上記(21)に記載の画像スタンピング方法。

(23) (a) 1ソース画素と、ソース画像内の、誤差値を含む対応する修正画素とを識別するステップと、

(b) 誤差値を含む修正画素の値と、スタンプされたソース画像内の対応する画素の値との間の差として定義される修正画素誤差の量を決定するステップと、(c) 隣接画素値の調整の総量が修正画素誤差と同等になるように、各隣接ソース画素の対応する修正画素が変更され、修正画素値が、スタンピング情報を組み込まれ、隣接する変更される画素値が、所望のスタンピング情報をまだ組み込まれておらず、後続のスタンピング処理で適当なスタンピング情報を組み込むためにさらに修正されることを特徴とする、連続する隣接ソース画素値を、識別されたソース画素から離れるにつれて量が減少する修正画素誤差を用いて調整することによって、修正画素誤差を誤差拡散するステップとを含み、スタンプされた画像が、複数のソース画素値によって定義され、対応する修正画素を形成するためにそれぞれがスタンピング情報を用いてスタンプされるソース画像であることを特徴とする、誤差拡散されたスタンプされた画像を提供する方法。

(24) (a) 誤差値を含む1ソース画素値と、スタンプされた画像内の対応する修正画素値とを識別するステップと、

(b) 誤差値を含むソース画素値と、対応する修正画素値との間の差として定義される修正画素誤差の量を決定するステップと、

(c) (i) 変更される修正画素のそれぞれと(ii) 修正画素とが、修正画素値としてスタンピング情報のうちの実質的に同様の部分を有するように、各隣接ソース

画素の対応する修正画素を変更することによって、連続する隣接ソース画素値を、識別されたソース画素から離れるにつれて量が減少する修正画素誤差を用いて調整することによって、修正画素誤差を誤差拡散するステップとを含み、スタンプされた画像が、複数のソース画素値によって定義され、対応する修正画素を形成するためにそれぞれがスタンピング情報を用いてスタンプされるソース画像であることを特徴とする、誤差拡散されたスタンプされた画像を提供する方法。

(25) (d) 誤差拡散されたスタンプされた画像を提供するため、全ソース画素が識別され、誤差拡散されるまで、ステップ(a)ないし(d)を反復するステップをさらに含むことを特徴とする、上記(24)に記載の誤差拡散されたスタンプされた画像を提供する方法。

(26) 検証キーに基づいて少なくとも1つの参照テーブル(LUT)を生成するための手段と、少なくとも1つのLUTに対応する写像関数を各ソース画像画素に適用し、写像関数の出力値が複数のスタンピング値のうちの対応する1つに対応するように各ソース画素を変更するための手段とを含む、ソース画像が複数のソース画像画素によって表され、スタンピング情報が複数のスタンピング値であることを特徴とする、ソース画像をスタンピング情報を用いてスタンプするためのシステム。

(27) 検証キーに基づいて少なくとも1つの参照テーブル(LUT)を生成するためのテーブル生成手段と、写像関数の出力値が複数のスタンピング値のうちの対応する1つを抽出するように、少なくとも1つのLUTに対応する写像関数を各スタンプされたソース画像画素に適用するための手段とを含む、スタンプされたソース画像が複数のスタンプされたソース画像画素によって表され、スタンピング情報が複数のスタンピング値であることを特徴とする、スタンピング情報を有するスタンプされたソース画像からスタンピング情報を抽出するためのシステム。

(28) 検証キーが、種値を含み、テーブル生成手段が、種値を使用する乱数ジェネレータを使用することによって、少なくとも1つのLUTの各項目を生成することとを特徴とする、上記(26)に記載のスタンプされたソース画像からスタンピング情報を抽出するためのシステム。

【図面の簡単な説明】

【図1】本発明の実施例に従う使用に適した、不可視画素\*

\* 像スタンピング処理と、画像及び検証キーの記憶配布処理の構成要素を含むシステムのブロック図である。

【図2】スタンプされた画像及び検証キーのアクセス及び配布を制御するサーバ内の、画像検証処理システムの実施例の機能ブロックを示すブロック図である。

【図3】スタンプされた画像及び検証キーを受け取るクライアント・コンピュータ・システム内の、画像検証処理システムの実施例の機能ブロックを示すブロック図である。

10 【図4】画像検証処理で、スタンプされた画像から埋め込まれたスタンピング情報を抽出する方法を示す流れ図である。

【図5】不可視画像スタンピング処理の実施例を示す、機能レベルのブロック図である。

【図6】画像の画素に対する不可視画像スタンピング処理の詳細を示す流れ図である。

【図7】画素値の修正と誤差拡散の処理の方法の詳細を示す流れ図である。

20 【図8】JPEG圧縮形式のソース画像に対する不可視画像スタンピング処理を示す、機能レベルのブロック図である。

【図9】JPEG圧縮形式のスタンプされた画像に対する画像検証処理を示す、機能レベルのブロック図である。

【図10】入力画像画素からの透かし値の抽出を示すブロック図である。

【図11】サンプル画像の例を示す図である。

【図12】図11に示された画像の直流係数画像の例である。

30 【符号の説明】

100 スタンピング情報

101 ソース画像

102 デジタル・コンピュータ

104 スタンプされたソース画像

105 検証キー

106 画像アーカイブ

107 保護記憶装置

108 画像サーバ

109 コンピュータ・ネットワーク

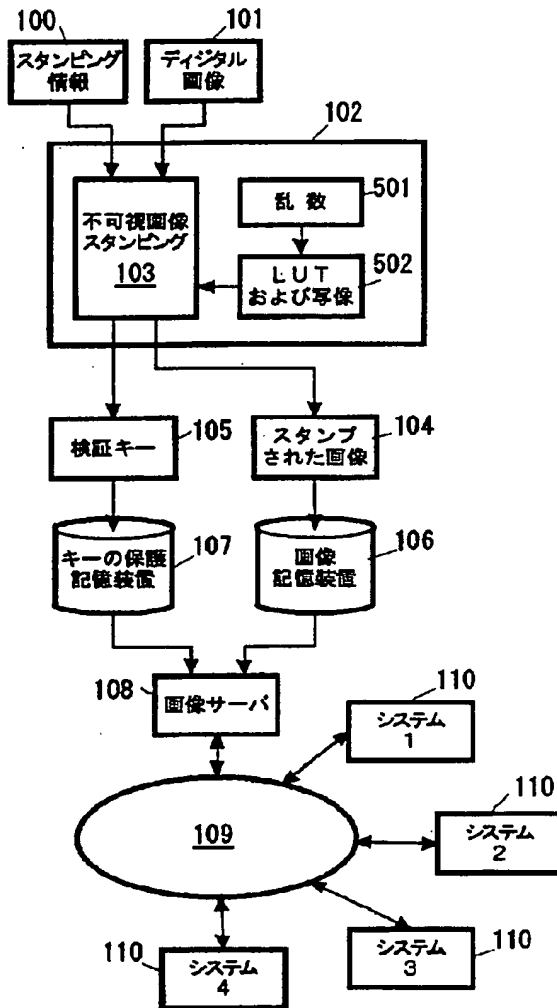
40 110 クライアント・コンピュータ・システム

501 乱数ジェネレータ

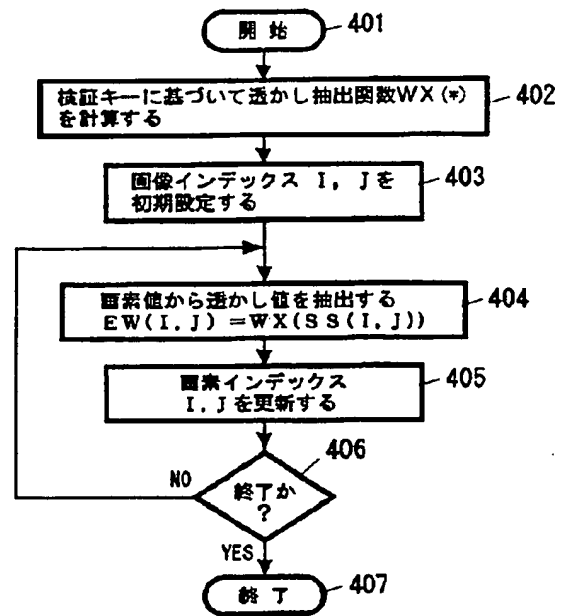
【図12】



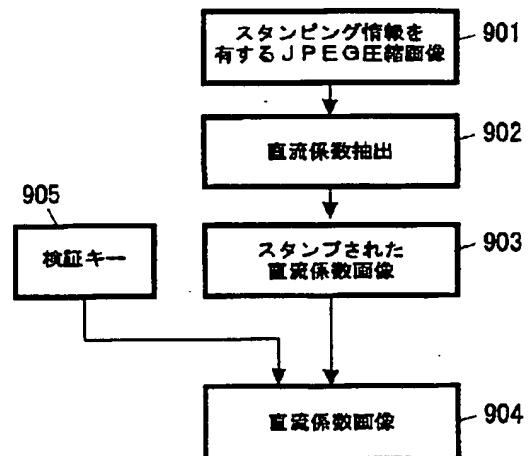
【図1】



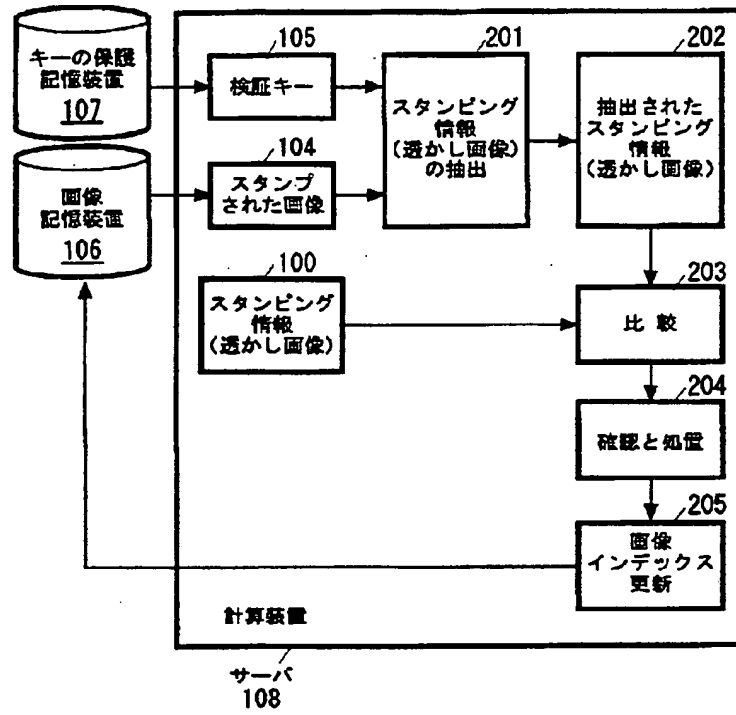
【図4】



【図9】



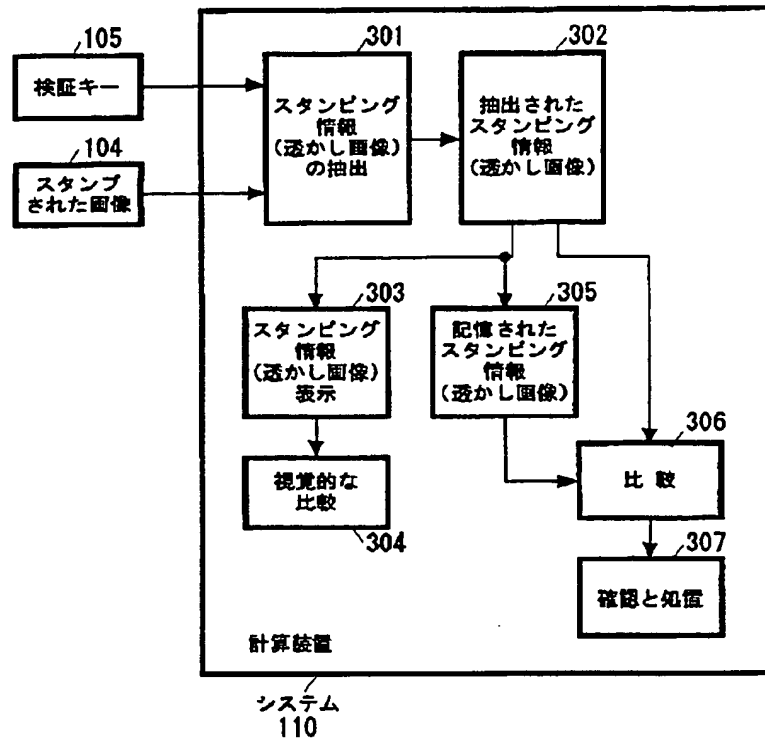
【図2】



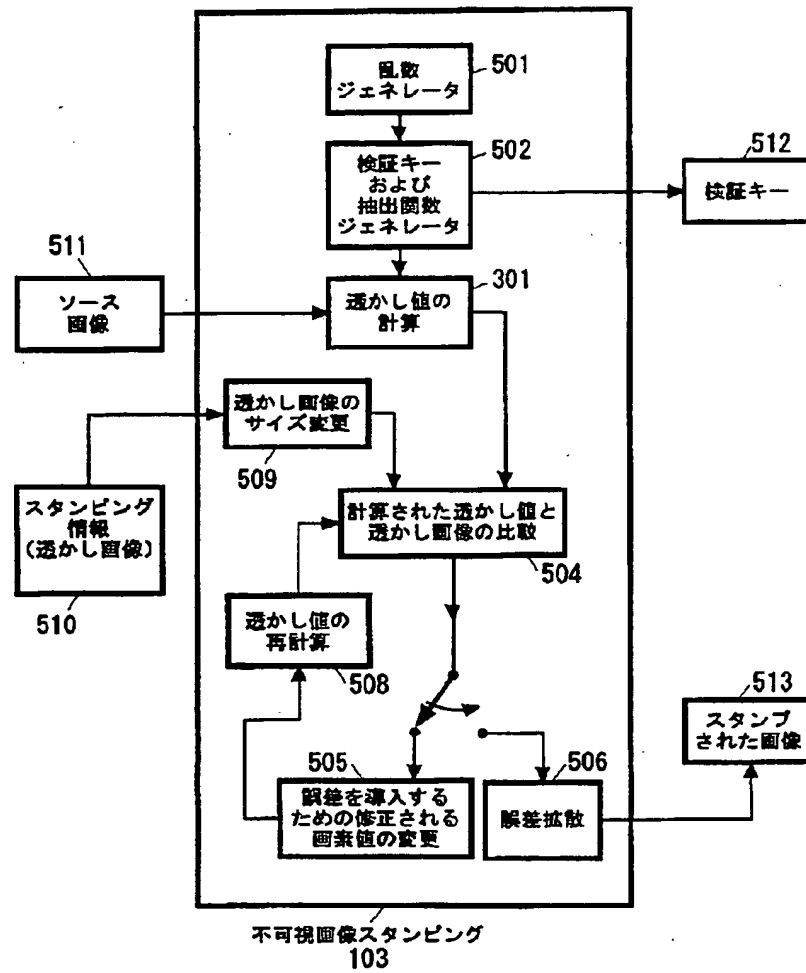
【図11】



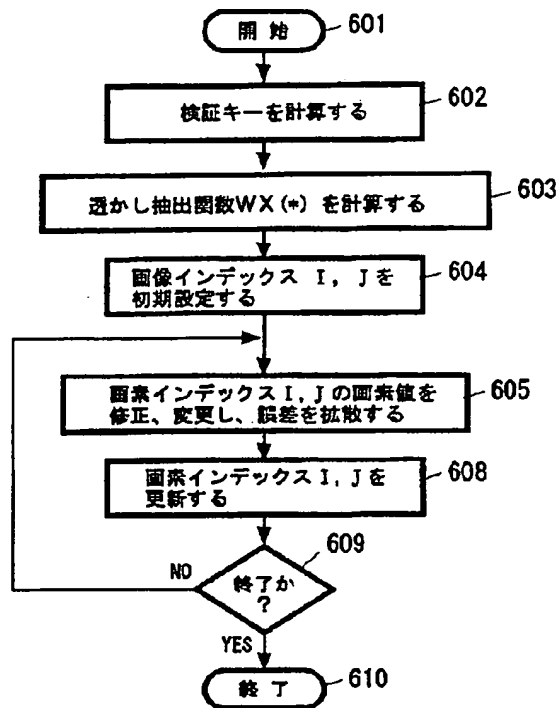
【図3】



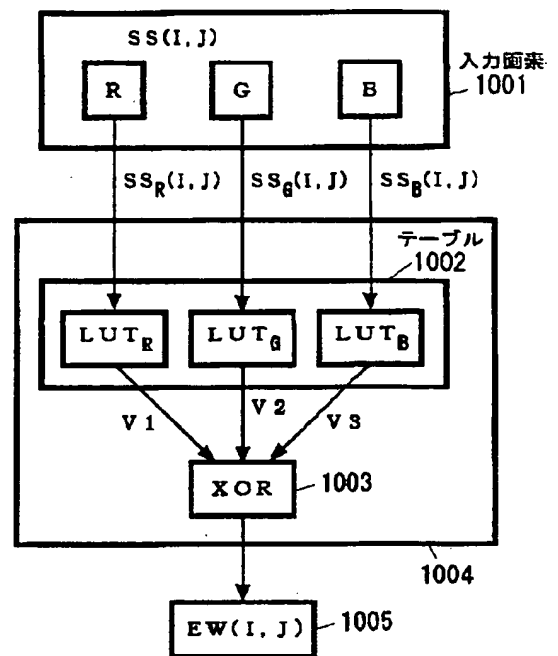
【図5】



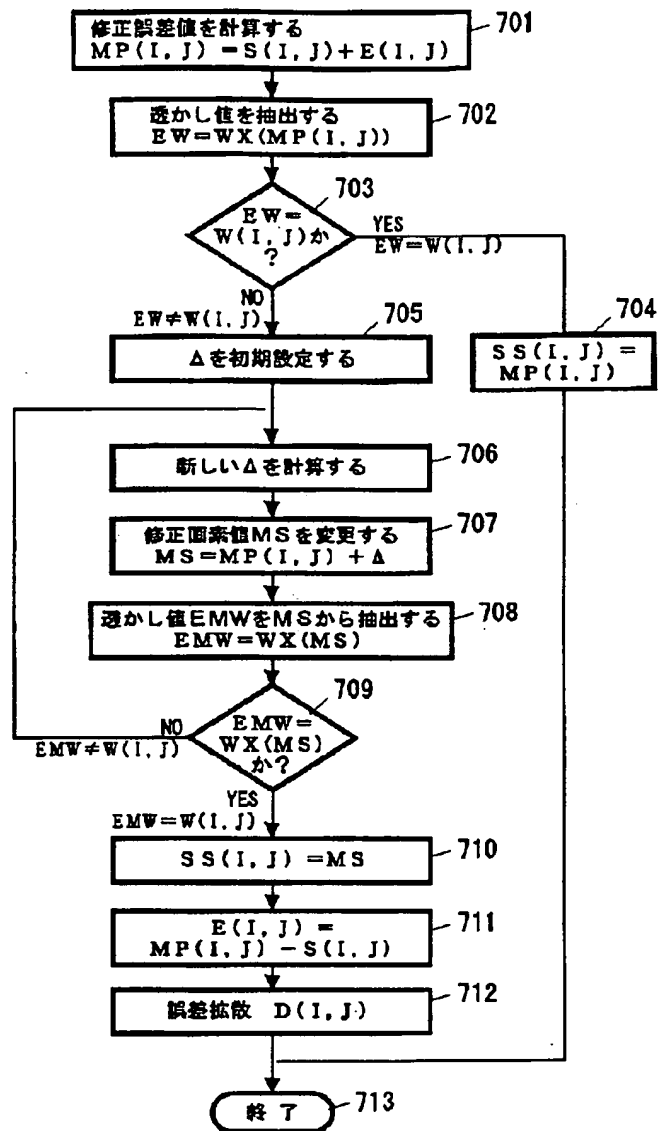
【図6】



【図10】

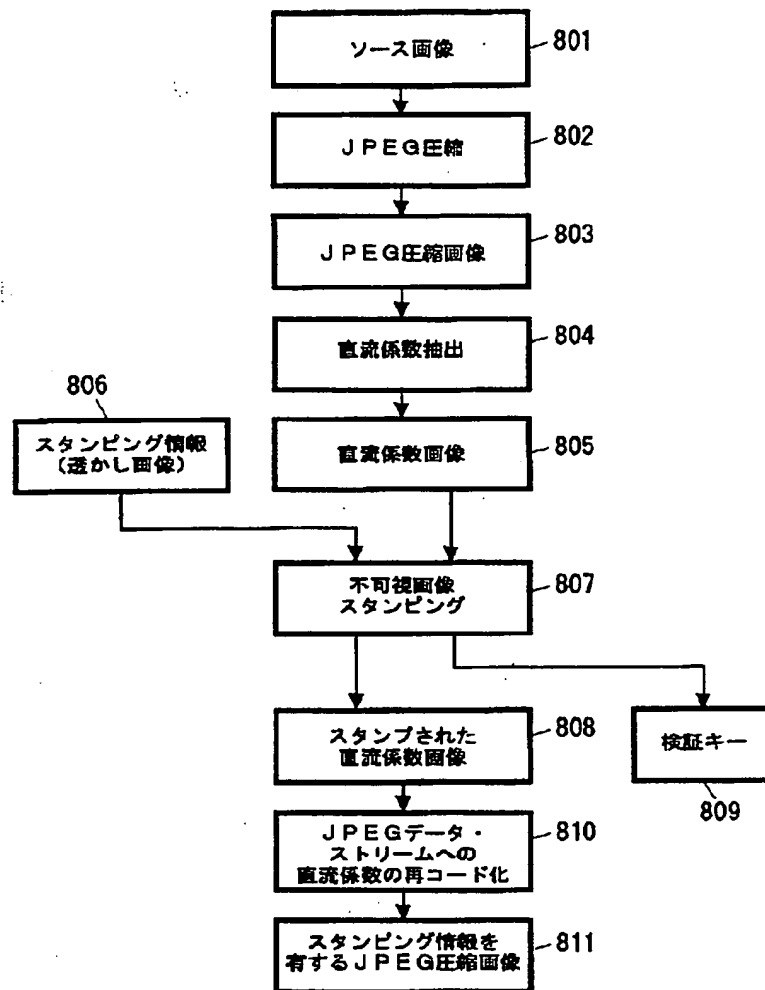


〔図7〕





【図8】



---

フロントページの続き

(72)発明者 マナーガ・ミン＝イー・ユン  
アメリカ合衆国10598 ニューヨーク州ヨ  
ークタウン・ハイツ ハイツ・ドライブ  
1537